Remarks

Claims 1-3, 5-14, 16-25, and 27-32 are pending in this application. Claims 4, 15, and 26 have been deleted without prejudice. Reconsideration of the application is respectfully requested in view of the following remarks.

The Office has asserted a rejection of claims 1-3, 5, 7-14, 16, 18-25, 27, and 29-32 under 35 USC §102(e) as anticipated by Leiba, U.S. Patent No. 6,260,065 ("Leiba") in an Office action dated February 14, 2006 (Office action). The Office has also asserted a rejection of claims 4, 6, 15, 17, 26, and 28 under 35 USC §103(a) as obvious over Leiba in view of Willis, U.S. Patent No. 6,321,376 ("Willis") in the same Office action. Applicants disagree with the rejections given and the Examiner's characterizations. Applicants address these rejections below.

1. With the goal of reaching a shared understanding of the disclosure of Leiba and Willis, the Applicants respectfully make the following observations.

A. Leiba

Leiba describes a method for testing a server to ensure that it complies with a standard, such as an RFC standard. [Leiba, 3:58-61.] The method includes providing test commands, feeding the test commands to the server being tested, and analyzing the server responses. [Leiba, 4:1-7.] These responses to a "given protocol command" can be received in non-deterministic order which "makes the test data usable with various implementations of a protocol, when such ordering may vary." [Leiba, 7:30-33.] Responses being received in non-deterministic order is further described in FIG. 6, and Leiba 7:8-32. Rather than just allowing a right or a wrong answer for a given response from a server, a response can be of several types: mandatory, optional, forbidden, several mandatory responses, and several optional responses (Leiba 2:45-47) allowing "server responses [to be expressed] in a flexible and general format...." [Leiba, 7:43.]

As shown in figure 1, "client test data is processed by a test engine and then transmitted to a server. The server responses are then processed to create a test result." [Leiba, 4:52-54.] The test data consists of two files: a test configuration file, which consists of server configuration information and test information, and a test input file, which consists of protocol commands and test elements. The software used to create the server itself under test is not used to generate test

data; rather, the server is treated as a black box consisting of inputs and outputs only, the outputs of which are used to determine if its response to the test data is correct.

B. Willis

Willis describes generating and applying language conformance tests to tools under test. [Willis, 3:18-24.] It provides a compiler which "translates the generator-oriented language specification into an intermediate (such as a C or C++ program) which can subsequently be compiled as an executable." [Willis, 6:36-39.] An "associated program language compiler" may be used to compile the generator-oriented language specification into the executable. [Willis, 7:60-67.]

2. Patentability over Leiba

The Office has asserted a rejection of claims 1-3, 5, 7-14, 16, 18-25, 27, and 29-32 under 35 USC §102(e) as anticipated by Leiba, U.S. Patent No. 5,883,661 ("Leiba"). Applicants respectfully traverse. For a 102(e) rejection to be proper, the cited art must show each and every element as set forth in a claim. (*See* MPEP § 2131.01.) However, the cited art does not so show. For example, with respect to claim 7, Leiba does not teach or suggest a method for "an identification of a mandatory call."

A. Claim 7

Applicants respectfully submit that Leiba fails to anticipate claim 7, because Leiba fails to teach or suggest "the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of the computer system." Specifically, claim 7 recites:

> A computer-implemented method of conformance-testing a software implementation with a software specification, the method comprising:
>
> producing a software object organized such that a step of the software specification and a corresponding code section of the software implementation are integrated in the software object; and
>
> the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a mandatory call comprised by the software specification to be stored in a memory of

the computer system.

For example, the Application states,

> If *calls to methods in other classes* (henceforth 'mandatory calls') are indicated in the specification, a check is made to confirm that those calls are made by the implementation. A check is made during each mandatory call to confirm that the state of the implementation remains in conformance with the specification. [Application, Page 20, lines 19-23.][Emphasis added.]

Leiba fails to show the feature of "mandatory calls." In its rejection of claim 7, the Action relies on various passages in Leiba; however, these passages describe a scenario involving a "mandatory response", not a "mandatory call" as recited in claim 7.

For example, the examiner quotes the passage below to support the rejection:

> The expected responses may be marked as one of a mandatory response, an optional response, a forbidden response, several mandatory responses and several optional responses for permitting testing of various types of responses, various types of test data and varied implementations of the server application. [Leiba, col. 2, lines 44-49.]

A "mandatory response" is one of the expected responses from a test sequence, as is made clear by the preceding sentence to the Leiba quote above: "The test sequence may be based on probabilistic transitions," [Leiba, col. 2, lines 43-44] and the discussion on Leiba, at 2A, above; not a "mandatory call", i.e., a call to a method or methods in other classes.

At least for this reason, Leiba fails to teach or suggest at least one element of claim 7. Claim 7 is thus in condition for allowance.

B. Claims 8-11

Claims 8-11 depend from claim 7. Since they depend from an allowable claims, they should be allowed for at least the reasons stated for claim 1. In view of the foregoing discussion of claim 1, the merits of the separate patentability of dependent claims 2-6 are not belabored at this time. Claims 2-6 should be allowable. Such action is respectfully requested.

C. Claims 18-22, 29-31

Independent claim 18 recites the following language:

> the software object comprising at least one instruction which, when executed by a computer system, causes an identification of a *mandatory call* comprised by the software specification to be stored in a memory of the computer system. [emphasis added.]

Claim 29 recites the following language:

> the software object comprising at least one instruction which, when executed by the computer system, causes a test that the state of the conformance-test enabled implementation conforms to the software specification during the *mandatory call.* [emphasis added.]

These claims both recite language related to the feature of a "mandatory call" which is not present in Leiba, as described above, with reference to the discussion concerning claim 7. In view of the foregoing discussion of claim 7, the merits of the separate patentability of claims 18 and 29 are not belabored at this time. Claims 18 and 29 should be allowable. Such action is respectfully requested. Furthermore, claims 19-22 depend from claim 18, and as the dependent claims of an allowable claim, should also be allowable. Similarly, claims 30-31 depend from claim 29, and as the dependent claims of an allowable claim, should be allowable themselves. Such action is respectfully requested.

3. ***Patentability over Leiba in view of Willis***

The Office has also asserted a rejection of claims 4, 6, 15, 17, 26, and 28 under 35 USC §103(a) as obvious over Leiba in view of Willis, U.S. Patent No. 6,321,376 ("Willis"). Applicants respectfully traverse. Leiba and Willis, taken separately or in combination, fail to teach or suggest at least one limitation of claims 4, 6, 15, 17, 26, and 28. Applicants respectfully submit the claims in their present form are allowable over the cited art.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim

limitations. (MPEP § 2142.)

Motivations to combine or modify references must come from the references themselves or be within the body of knowledge in the art. (*See* MPEP § 2143.01.)

A. Claim 1

Amended claim 1 now incorporates all of the limitations of claim 4, and so, is argued here. Claim 1 has also been amended to add the words "source code" in the following phrase: "compiling the software implementation source code." Support for this change in the specification can be found at FIG. 1 in box 102, which reads "implementation source code."

Leiba and Willis, either together or in combination fail to teach or suggest "compiling the software implementation source code from a first high-level language into an intermediate language; compiling the software specification from a second high-level language into the intermediate language; and producing the conformance-test enabled implementation in the intermediate language" as recited in amended Claim 1.

The examiner states that "Leiba does not explicitly disclose compiling the software implementation from a first high-level language into an intermediate language; compiling the software specification from a second high-level language into the intermediate language; and producing the conformance-test enabled implementation in the intermediate language" [Office action, p. 10.] We agree. However, Willis also does not teach the language recited from amended claim 1 above. For example, to teach or suggest the claim limitation "compiling the software implementation source code from a first high-level language into an intermediate language; compiling the software specification from a second high-level language into the intermediate language; and producing the conformance-test enabled implementation in the intermediate language" the Action relies on the language, below:

> In this embodiment, the formal language specification is parsed into a specification intermediate (Block 31), then a test case compiler creates a compiled test case generator, potentially using an intermediate programming language manifestation of the generator and associated programming language compiler, (both within Block 50) to yield an executable generator. The resulting compiled generator (Block 51) produces test cases in a manner functionally equivalent to the interpretive generator (Block 32) discussed above in the context of FIG. 3.] [Willis, 7:59-8:4.]

Specifically, the language quoted above from Willis, at a minimum fails to teach or

suggest *software implementation source code*, let alone the language from claim 1 "compiling the software implementation source code from a first high-level language into an intermediary language". A software implementation is defined in the specification as follows: "A software implementation in the form of a class definition may be compiled into an intermediate language (IL) form." [Specification, p. 5, ll. 5-6.] "With reference to the system embodiment 100 of Figure 1, software source code 102 is compiled into an "intermediate language" (IL) form 104 before being loaded into a runtime environment 106.... The source code 102 is typically a high-level programming language form such as C++, C# (pronounced C sharp), Visual Basic, and so on." [Specification, p. 5, ll.12-16.]

The reference does describe a test case compiler, however, a test case compiler is quite different than "software implementation source code." One of ordinary skill in the art could not be expected to surmise the claimed arrangement of "compiling the software implementation source code from a first high-level language into an intermediary language" from the mention of a quite different feature—the test case compiler. For example, the test case compiler in Willis is described as follows:

> The test case compiler and generation environment embodiment comprises a compiler (Block 2) and a test case generation environment (Block 3). The compiler translates the generator-oriented language specification into an intermediate (such as a C or C++ program) which can subsequently be compiled into an executable. [Willis, 6:34-39.]

The test case compiler is described as having as input only the language specification, which it compiles into an intermediate program. It is not "software implementation source code" as required by claim 1. Thus, one of ordinary skill in the art could not be expected to surmise the claimed arrangement of "compiling the software implementation source code from a first high-level language into an intermediary language" from the mere mention of a different feature – a test case compiler—in Willis. Claim 1 should now be allowable. Such action is respectfully requested.

B. Claims 2-6

Claims 2-6 depend from claim 1. Since they depend from claim 1, they should be allowed for at least the reasons stated for claim 1. In view of the foregoing discussion of claim 1,

the merits of the separate patentability of dependent claims 2-6 are not belabored at this time. Claims 2-6 should be allowable. Such action is respectfully requested.

C. Claims 12-14, 16-17

Claim 12 has been amended to include all of the language of claim 15. Claim 12 now includes the language:

> compiling the software implementation from a first high-level language into an intermediate language;
> compiling the software specification from a second high-level language into the intermediate language; and
> producing the conformance-test enabled implementation in the intermediate language

In light of the discussion in section 3A discussing claim 1, claim 12 is also patentable, in similar grounds, which will not be belabored here. Claims 13-14, and 16- 17 depend from claim 12 and include all of the language of claim 12. As noted in section 3A, Leiba fails to teach or suggest the above-cited language of claim 12. Willis also fails to teach or suggest the above-cited language of claim 12. Therefore, claim 12 is now allowable.

Claims 13-14, and 16-17 depend from claim 12 and include all of the language of claim 12. As they depend from an allowable claim, they are themselves allowable. Such action is respectfully requested.

D. Claims 23-25, 27-28

Claim 23 has been amended to include all of the language of claim 26. Claim 23 now includes the language:

> compiling the software implementation from a first high-level language into an intermediate language;
> compiling the software specification from a second high-level language into the intermediate language; and
> producing the conformance-test enabled implementation in the intermediate language.

In light of the discussion in section 3A discussing claim 1, claim 23 is also patentable, in similar grounds, which will not be belabored here. Willis also fails to teach or suggest the above-cited

language of claim 23. Therefore, claim 23 is now allowable. Claims 24-25 and 27-28 depend from claim 23 and include all of the language of claim 23. As they depend from an allowable claim, they are themselves allowable. Such action is respectfully requested.

4. Request For Interview

If any issues remain in light of these remarks and amendments, the Examiner is formally requested to contact the undersigned attorney to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the preceding formal Amendment and the above remarks so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

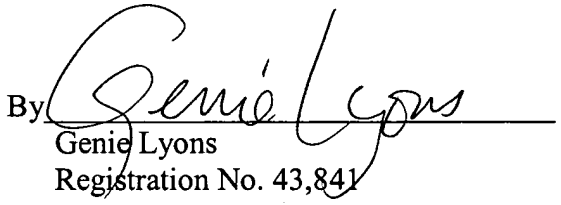
Conclusion

The claims in their present form should now be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By 
Genie Lyons
Registration No. 43,841